AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of encrypting a data string, comprising:

receiving, from a user, the data string, as plaintext;

generating an n-dimensional entity, wherein the n-dimensional entity comprises random bits; and

for each bit in the data string:

reading a number of bits from the n-dimensional entity;

performing an action based in part on the read number of bits;

generating a bit sequence;

selecting a direction within the n-dimensional entity based in part on the generated bit sequence;

determining an offset between a cursor position and a match bit within the n-dimensional entity, wherein the match bit is based in part on the action, the direction, and the each bit in the data string; and

modifying the generated bit sequence with the determined offset by inserting the determined offset into the generated bit sequence to generate a row within an encoded data string corresponding to each bit in the data string; and

storing the resulting encoded data string as an encoded representation of the received plaintext data string.

2. (Original) The method of claim 1, wherein generating the n-dimensional entity further comprises:

generating a seed for a random number generator;

determining a number of dimensions of the n-dimensional entity;

determining a length for each dimension of the n-dimensional entity; and

populating the n-dimensional entity with bits from the random number generator.

3. (Original) The method of claim 2, wherein the number of dimensions is determined based in part on at least one of a user selectable input, a default value, and a random number.

4. (Original) The method of claim 2, wherein the length of each dimensions is determined based in part on at least one of a user selectable input, a default value, and a random number.

5. (Original) The method of claim 2, wherein the random number generator is arranged to produce a pseudo-random bit sequence.

6. (Original) The method of claim 1, wherein reading the number of bits from the n-dimensional entity further comprises reading a sequence of bits equal to a size of an op-code.

7. (Original) The method of claim 6, wherein the size of the op-code is selected from at least one of a default size and a user selectable input.

8. (Original) The method of claim 1, wherein performing the action further comprises a means for associating an action to the read number of bits.

9. (Original) The method of claim 1, wherein performing the action further comprises:
interpreting the read number of bits as an op-code;
determining an action associated with the op-code; and
executing the action associated with the op-code.

10. (Original) The method of claim 1, wherein performing the action further comprises associating the read number of bits with an action using at least one of a database, a table, a linked-list, and a program.

11. (Original) The method of claim 1, wherein performing the action further comprises performing at least one of changing a cursor position, switching a bit state, reading a bit, modifying a bit, generating another n-dimensional entity, changing a direction, and modifying an interpretation of a bit state.

12. (Original) The method of claim 1, wherein generating a bit sequence further comprises generating a truly random bit sequence.

13. (Original) The method of claim 1, further comprising, combining an encoded data string associated with a bit in the data string with another encoded data string associated with a different bit in the data string.

14. (Original) The method of claim 1, further comprising exclusive or-ing each encoded data string with a previous encoded data string, wherein a first encoded data string is exclusively or-ed with a last encoded data string.

15. (Original) The method of claim 1, further comprising combining bits within an encoded data string with a corresponding bit within an obfuscation table.

16. (Original) The method of claim 1, further comprising modifying the length of at least one encoded data string.

17. (Original) The method of claim 1, further comprising including at least one random data string with the each encoded data string.

18. (Original) The method of claim 1, wherein generating the n-dimensional entity further comprises determining a fingerprint associated with a computing system in which the method operates.

19. (Currently Amended) A method of encrypting a data string, comprising:

receiving, from a user, the data string to be encoded;

generating an n-dimensional entity, wherein the n-dimensional entity is populated with pseudo-random bits;

for each bit in the data string:

determining a cursor position within the n-dimensional entity;

determining a direction within the n-dimensional entity;

determining a number of bits in the n-dimensional entity, wherein the bits are read from the determined cursor position along the determined direction;

performing an action based in part on the determined number of bits;

generating a bit sequence;

selecting another direction based in part on the bit sequence;

determining an offset between a match bit within the n-dimensional entity and the cursor position, wherein the match bit is based in part on the action, the other direction, and the each bit in the data string; and

modifying the bit sequence with the determined offset by inserting the determined offset into the generated bit sequence to generate an encoded data string for each bit in the data string; and

storing the encoded data string in a data store as an encoded representation of the data string..

20. (Original) The method of claim 19, wherein generating the bit sequence further comprises generating a truly random bit sequence.

21. (Original) The method of claim 19, wherein determining the number of bits in the n-dimensional entity, further comprises a means for determining an action based in part on the read number of bits.

22. (Original) The method of claim 19, wherein performing an action further comprises:

interpreting the determined number of bits as an op-code; and

executing an action associated with the op-code.

23. (Original) The method of claim 19, further comprising employing an obfuscation table to obfuscate the encoded data string for each bit in the data string.

24. (Original) The method of claim 19, wherein determining the cursor position further comprises:

receiving a cursor position; and

normalizing the received cursor position to within a boundary of the n-dimensional entity.

25. (Original) The method of claim 24, wherein normalizing the received cursor position further comprises employing a circular orbiting algorithm to the cursor position until the cursor position is within the boundary of the n-dimensional entity.

26. (Original) The method of claim 19, wherein selecting another direction further comprises employing a predetermined set of bits in the bit sequence to select the other direction.

27. (Original) The method of claim 19, wherein modifying the bit sequence with the offset further comprises overwriting a predetermined set of bits in the bit sequence with the determined offset.

28. (Original) The method of claim 19, wherein determining the offset further comprises generating another n-dimensional entity, if the match bit is not located.

29. (Original) The method of claim 19, wherein determining the offset further comprises setting a bit in the bit sequence, if the match bit is not located.

30. (Original) The method of claim 19, wherein generating the n-dimensional entity, further comprises:

generating a fingerprint based in part on a computing system in which the method operates; and

determining a characteristic of the n-dimensional entity based in part on the fingerprint.

31. (Original) The method of claim 30, wherein the characteristic of n-dimensional entity further comprises at least one of a length of a side, a number of dimensions, and a seed for a random number generator which is enabled to populate the n-dimensional entity with random bits.

32. (Original) The method of claim 30, wherein the fingerprint further comprises a hash of at least one of a Central Processing Unit's (CPU's) kernel speed, CPU serial number, CPU family identity, CPU manufacturer, an operating system globally unique identifier (QUID), a hardware component enumeration, Internet Protocol (IP) address, BIOS serial number, disk serial number, kernel version number, operating system version number, operating system build number, machine name, installed memory characteristic, physical port enumeration, customer supplied ID, and a MAC address.

33. (Original) The method of claim 32, wherein the hash further comprises at least one a Message Digests (MD), a secure hash, and a Secure Hash Algorithm (SHA).

34. (Original) The method of claim 19, wherein generating the n-dimensional entity, further comprises:

creating a digest in part from a fingerprint associated with a computing system in which the method operates;

seeding a pseudo-random number generator in part with the digest;

determining a number of dimensions of the n-dimensional entity based in part on an output of the pseudo-random number generator; and

determining a length of a side of the n-dimensional entity based in part on another output of the pseudo-random number generator.

35. (Original) The method of claim 34, wherein creating the digest further comprises, combining the fingerprint with a user seed to create the digest.

36. (Currently Amended) A system for encrypting a data string, comprising:

an entity generator that is arranged to generate an n-dimensional entity;

a mapper, arranged to receive the n-dimensional entity, and perform actions, comprising:

receiving a data string; and

for each bit in the data string:

reading a number of bits from the n-dimensional entity;

performing an action based in part on the read number of bits;

generating a bit sequence;

selecting a direction within the n-dimensional entity based in part on the generated bit sequence;

determining an offset between a cursor position and a match bit within the n-dimensional entity, wherein the match bit is based in part on the action, the direction, and the each bit in the data string; and

modifying the generated bit sequence by inserting the generated bit sequence into [[with]] the determined offset to generate an encoded data string; and

storing the encoded data string in a data store as an encoded representation of the received data string.

37. (Original) The system of claim 36, wherein the entity generator generates the n-dimensional entity by performing actions, comprising:

determining a seed for a random number generator;

determining a number of dimensions of the n-dimensional entity;

determining a length for each dimension of the n-dimensional entity;

populating the n-dimensional entity with bits from the random number generator; and

determining an initial cursor position within the n-dimensional entity.

38. (Original) The system of claim 37, wherein determining the seed further comprises creating the seed from a combination of a user seed and a fingerprint that is associated with a computing system in which the system operates.

39. (Original) The system of claim 37, wherein the initial cursor position is determined based in part on normalizing a received cursor position to within a boundary of the n-dimensional entity.

40. (Original) The system of claim 37, wherein the number of dimensions is determined based in part on at least one of a user selectable input, a default value, and a random number.

41. (Original) The system of claim 36, wherein the generated n-dimensional entity is populated with pseudo-random bits.

42. (Original) The system of claim 36, wherein performing the action further comprises performing at least one of changing a cursor position, switching a bit state, reading a bit, generating another n-dimensional entity, changing a direction, and modifying an interpretation of a bit state.

43. (Original) The system of claim 36, wherein generating a bit sequence further comprises generating a truly random bit sequence.

44. (Currently Amended) An apparatus for encrypting a data string, comprising:

a transceiver that receives the data string and sends an encoded array to another apparatus; and

coupled to the transceiver, an n-dimensional encrypter that is arranged to perform actions, comprising:

generating an n-dimensional entity, wherein the n-dimensional entity comprises random bits; and

for each bit in the received data string:

reading a number of bits from the n-dimensional entity;

performing an action associated with the read number of bits;

generating a bit sequence;

selecting a direction within the n-dimensional entity based in part on the generated bit sequence;

determining an offset between a cursor position and a match bit within the n-dimensional entity, wherein the match bit is based in part on the action, the direction, and the each bit in the received data string; and

modifying the generated bit sequence with the determined offset by inserting the determined offset into the generated bit sequence to generate an

encoded data string, wherein the encoded data string represents a row within the encoded array.

45. (Original) The apparatus of claim 44, wherein reading the number of bits from the n-dimensional entity further comprises reading a sequence of bits equal to a size of an op-code.

46. (Original) The apparatus of claim 44, wherein performing the action further comprises performing at least one of changing a cursor position, switching a bit state, reading a bit, modifying a bit, generating another n-dimensional entity, changing a direction, and modifying an interpretation of a bit state.

47. (Currently Amended) An apparatus of encrypting a data string, comprising:

a means for generating an n-dimensional entity;

a means for receiving the data string as input from a user;

a means for performing an action for each bit in the data string based in part on the n-dimensional entity;

a means for generating a random bit sequence associated with each bit in the data string; and

a means for modifying the each random bit sequence [[with]] by inserting into each random bit sequence an offset associated with each bit in the data string, wherein the offset is based in part on the action, the n-dimensional entity, and the each bit in the data string; and

means for storing the modified bit sequence in a storage device.